

**AMENDMENTS TO THE CLAIMS:**

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A method for moving data objects in a computer system from a first storage location to a second storage location, the method comprising:

[[a))] selecting ~~one or more~~ a data object ~~objects~~ from the first storage location;  
[[b))] assigning ~~at least one~~ an identifier (ID) ~~of at least one type~~ to each of the selected data objects object;

[[c))] storing the ID in a transactional type lock object;

[[d))] determining whether the ID is stored successfully in the transactional type lock object, and upon a successful storage, storing the ID in a permanent type lock object;

[[e))] determining whether the ID is stored successfully in the permanent type lock object, and upon a successful storage, deleting the ID from the transactional type lock object;

[[f))] storing the data object, ~~the ID of which is contained in the permanent type lock object,~~ at the second storage location; and

assigning the second storage location to the ID in the permanent type lock object;

[[g)]] deleting the data object, ~~the ID of which is contained in the permanent type lock object,~~ from the first storage location; and

[[h)]] deleting the ID from the permanent type lock object after the respective data object assigned to that ID has been deleted from the first storage location.

2. (Currently Amended) The method of claim 1, wherein ~~each~~ the data object comprises one or more fields of one or more tables, and wherein the ID comprises one or more key fields of the one or more tables.

3. (Currently Amended) The method of claim 1, wherein the data object is stored in ~~one or more files~~ a file and wherein an assignment of the ID to ~~a filename~~ the file or a name of the file, in which the data object assigned to the ID is to be stored, is stored in the permanent type lock object.

4. (Currently Amended) The method of claim 1, wherein the ID is stored in the transactional type lock object after ~~the step of assigning at least one identifier of at least one type~~ the ID to each of the selected data objects ~~for the respective data object~~.

5. (Currently Amended) The method of claim 1, wherein storing ~~said~~ the ID in the permanent type lock object ~~further~~ comprises storing ~~the ID~~ IDs of all selected data objects in the permanent type lock object before ~~a first storing process according to step f) is started~~ any of the selected data objects at the second storage location.

6. (Currently Amended) The method of claim 1, further comprising:

[[i)]] checking whether [[an]] the ID for one of the selected data object objects has been stored in [[a]] at least one of the transactional type lock object and the permanent type lock object, and if the ID has been stored, skipping ~~at least step f) for~~ the one of storing the selected data objects object at the second storage location.

7. (Currently Amended) The method of claim 1, further comprising:

[[j)]] checking whether the data object is contained in the second storage location and if the data object is contained, skipping ~~at least step f) for that~~ storing the data object at the second storage location.

8. (Currently Amended) The method of claim 7, wherein ~~said~~ the checking comprises is performed by querying at least one of the transactional type lock object and the permanent type lock object.

9. (Currently Amended) The method of claim 1, further comprising:

[[k)]] determining whether the ID was successfully stored in the transactional type lock object, and upon an unsuccessful storage, checking whether the data object assigned to the respective ID has been ~~completely~~ stored in the second storage location, and if the ~~respective ID~~ data object has not been ~~completely~~ stored, skipping at

~~least steps g) and h) for that~~ deleting the data object from the first storage location and skipping deleting the ID from the permanent type lock object.

10. (Previously Presented) The method of claim 1 for use in an enterprise resource planning software.

11. (Currently Amended) A computer system for processing data, the computer system comprising:

memory means for storing program instructions;

input means for entering the data;

storage means for storing the data;

a processor responsive to the program instructions, wherein the program instructions comprise program code means for performing a method for moving data objects in the computer system from a first storage location to a second storage location, the method comprising:

selecting ~~one or more~~ a data object objects from the first storage location;

assigning ~~at least one~~ an identifier (ID) ~~of at least one type~~ to each of the ~~selected data objects~~ object;

storing the ID in a transactional type lock object;

determining whether the ID is stored successfully in the transactional type lock object, and upon a successful storage, storing the ID in a permanent type lock object;

determining whether the ID is stored successfully in the permanent type lock object, and upon a successful storage, deleting the ID from the transactional type lock object;

storing the data object, ~~the ID of which is contained in the permanent type lock object,~~ at the second storage location; and

assigning the second storage location to the ID in the permanent type lock object;

deleting the data object, ~~the ID of which is contained in the permanent type lock object,~~ from the first storage location; and

deleting the ID from the permanent type lock object after the respective data object assigned to that ID has been deleted from the first storage location.

12. (Currently Amended) A computer readable medium comprising instructions for performing a method for moving data objects in a computer system from a first storage location to a second storage location, the method comprising:

selecting ~~one or more~~ a data object objects from the first storage location;

assigning ~~at least one~~ an identifier (ID) of ~~at least one type~~ to each of the selected data objects object;

storing the ID in a transactional type lock object;

determining whether the ID is stored successfully in the transactional type lock object, and upon a successful storage, storing the ID in a permanent type lock object;

determining whether the ID is stored successfully in the permanent type lock object, and upon a successful storage, deleting the ID from the transactional type lock object;

storing the data object, ~~the ID of which is contained in the permanent type lock object,~~ at the second storage location; and

assigning the second storage location to the ID in the permanent type lock object;

deleting the data object, ~~the ID of which is contained in the permanent type lock object,~~ from the first storage location; and

deleting the ID from the permanent type lock object after the respective data object assigned to that ID has been deleted from the first storage location.

13-14. (Cancelled).

15. (Currently Amended) The computer readable medium of claim 12, wherein ~~each~~ the data object comprises one or more fields of one or more tables, and wherein the ID comprises one or more key fields of the one or more tables.

16. (Currently Amended) The computer readable medium of claim 12, wherein the data object is stored in ~~one or more files~~ a file and wherein an assignment of the ID to ~~a filename~~ the file or a name of the file, in which the data object assigned to the ID is to be stored, is stored in the permanent type lock object.

17. (Currently Amended) The computer readable medium of claim 12, wherein the ID is stored in the transactional type lock object after ~~the step of assigning at least one identifier of at least one type~~ the ID to each of the selected data objects for ~~the respective data object~~.

18. (Currently Amended) The computer readable medium of claim 12, wherein storing said the ID in the permanent type lock object ~~further comprises~~ [[:]] storing ~~the ID~~ IDs of all selected data objects in the permanent type lock object before storing any of the selected data objects object at the second storage location.

19. (Currently Amended) The computer readable medium of claim 12, wherein the method further comprising comprises:

checking whether [[an]] the ID for one of the selected data object objects has been stored in [[a]] at least one of the transactional type lock object and the permanent type lock object, and if the ID has been stored, skipping the storing [[of]] the data object, ~~the ID of which is contained in the permanent type lock object,~~ at the second storage location ~~for the one of the selected data objects~~.

20. (Currently Amended) The computer readable medium of claim 12, wherein the method further comprising comprises:

checking whether the data object is contained in the second storage location and if the data object is contained, skipping the storing ~~[[of]]~~ the data object, ~~the ID of which is contained in the permanent type lock object,~~ at the second storage location for that data object.

21. (Currently Amended) The computer readable medium of claim ~~[[12]]~~ 20, wherein the checking ~~is performed by~~ comprises querying at least one of the transactional type lock object and the permanent type lock object.

22. (Currently Amended) The computer readable medium of claim 12, wherein the method further comprising ~~comprising~~:

determining whether the ID was successfully stored in the transactional type lock object, and upon an unsuccessful storage, checking whether the data object assigned to the respective ID has been ~~completely~~ stored in the second storage location, and if the ~~respective ID~~ data object has not been ~~completely~~ stored, skipping the deleting ~~[[of]]~~ the data object, ~~the ID of which is contained in the permanent type lock object,~~ from the first storage location, ~~and the deleting of the ID from the permanent type lock object after the respective data object assigned to that ID has been deleted from the first storage location for that data object and~~ skipping deleting the ID from the permanent type lock object.



23. (Currently Amended) A computer system for processing data, the computer system comprising:

means for selecting ~~one or more~~ a data object ~~objects~~ from the first storage location;

means for assigning ~~at least one~~ an identifier (ID) ~~of at least one type~~ to each of the ~~selected data objects~~ object;

means for storing the ID in a transactional type lock object;

means for determining whether the ID is stored successfully in the transactional type lock object, and upon a successful storage, storing the ID in a permanent type lock object;

means for determining whether the ID is stored successfully in the permanent type lock object, and upon a successful storage, deleting the ID from the transactional type lock object;

means for storing the data object, ~~the ID of which is contained in the permanent type lock object,~~ at the second storage location; and

means for assigning the second storage location to the ID in the permanent type lock object;

means for deleting the data object, ~~the ID of which is contained in the permanent type lock object,~~ from the first storage location; and

means for deleting the ID from the permanent type lock object after the respective data object assigned to that ID has been deleted from the first storage location.

24. (Currently Amended) The computer system of claim 23, wherein ~~each~~ the data object comprises one or more fields of one or more tables, and wherein the ID comprises one or more key fields of the one or more tables.

25. (Currently Amended) The computer system of claim 23, wherein ~~further~~ comprising:

means for storing the data object ~~is stored in one or more files~~ a file; and  
means for storing ~~wherein an assignment of the ID to a filename~~ the file or a  
name of the file, in which the data object assigned to the ID is to be stored, ~~is stored in~~  
the permanent type lock object.

26. (Currently Amended) The computer system of claim 23, wherein ~~further~~ comprising means for storing the ID ~~is stored in the transactional type lock object after~~  
~~the step of assigning at least one identifier of at least one type~~ the ID to each of the  
~~selected data objects for the respective data object.~~

27. (Currently Amended) The computer system of claim 23, wherein the  
means for storing ~~said~~ the ID in the permanent type lock object ~~further comprises~~ [[:]  
means for storing ~~the ID~~ IDs of all selected data objects in the permanent type lock  
object before storing any of the selected data objects ~~object~~ at the second storage  
location.

28. (Currently Amended) The computer system of claim 23, further comprising:

means for checking whether ~~[[an]]~~ the ID ~~for that~~ of the data object has been stored in ~~[[a]]~~ at least one of the transactional type lock object and the permanent type lock object, and if the ID has been stored, skipping ~~the storing~~ ~~[[of]]~~ the data object, ~~the ID of which is contained in the permanent type lock object~~, at the second storage location ~~for that data object~~.

29. (Currently Amended) The computer system of claim 23, further comprising:

means for checking whether the data object is contained in the second storage location and if the data object is contained, skipping ~~the storing~~ ~~[[of]]~~ the data object, ~~the ID of which is contained in the permanent type lock object~~, at the second storage location ~~for that data object~~.

30. (Currently Amended) The computer system of claim ~~[[23]]~~ 29, wherein the means for checking is performed by comprises means for querying at least one of the transactional type lock object and the permanent type lock object.

31. (Currently Amended) The computer system of claim 23, further comprising:

means for determining whether the ID was successfully stored in the transactional type lock object, and upon an unsuccessful storage, checking whether the data object assigned to the respective ID has been ~~completely~~ stored in the second storage location, and if the ~~respective ID~~ data object has not been ~~completely~~ stored, skipping the deleting ~~[[of]]~~ the data object, ~~the ID of which is contained in the permanent type lock object, from the first storage location, and the deleting of the ID from the permanent type lock object after the respective data object assigned to that ID has been deleted from the first storage location for that data object and~~ skipping deleting the ID from the permanent type lock object.